# Reti Neurali in grado di apprendere

*Una mente intelligente è quella che è in costante apprendimento.*

**Bruce Lee**

***Giorgio Buttazzo*** *Scuola Superiore Sant'Anna, Pisa*

**Grazie alle conoscenze acquisite sul cervello umano, l'intelligenza artificiale è riuscita a sviluppare modelli matematici del neurone biologico che oggi vengono utilizzati per costruire reti neurali artificiali, ossia sistemi computazionali in grado di apprendere dai propri errori. Negli anni sono stati sviluppati diversi paradigmi di apprendimento. Quello più noto è il paradigma supervisionato, che consente ad una rete di apprendere delle associazioni attraverso un insieme di esempi preparati da un trainer, che indica alla rete la risposta giusta per ognuno di essi. Un'altra modalità di apprendimento è quella basata su premi e punizioni, che presuppone l'esistenza di un `critico`, il quale, questa volta, non conosce le risposte giuste da suggerire alla rete, ma giudica solo la bontà delle azioni prodotte, penalizzando o incentivando la rete a produrle nuovamente. Infine, nel paradigma non supervisionato, la rete neurale è in grado di auto-organizzarsi in funzione unicamente dei dati che riceve in ingresso, specializzandosi al punto da discriminarli in base alle differenze più salienti rilevate.**

**L'obiettivo di questo articolo è di introdurre i concetti fondamentali del calcolo neuronale, presentando i vari meccanismi di apprendimento sviluppati negli anni, illustrandone alcuni esempi di utilizzo e le potenzialità future.**

## Introduzione

La capacità di apprendere dai propri errori e, in generale, di adattarsi ai cambiamenti è una caratteristica essenziale dell'intelligenza. Senza questa capacità, probabilmente la specie umana non sarebbe diventata la specie dominante sul pianeta Terra.

Nel campo dell'intelligenza artificiale, l'importanza dell'apprendimento è stata riconosciuta solo di recente, in quanto le metodologie classiche e gli algoritmi sviluppati tra gli anni sessanta e gli anni 2000, avevano prodotto ottimi risultati in diversi campi applicativi, quali le diagnosi mediche, le previsioni meteorologiche, la dimostrazione automatica di teoremi, la comprensione di testi e i giochi di strategia, come dama e scacchi. Tra i successi più noti, ricordiamo la sfida scacchi-

stica tra Deep Blue (un supercomputer dell'IBM) e il campione del mondo Gary Kasparov, conclusasi l'11 maggio del 1997 con la vittoria di Deep Blue per 3.5 a 2.5.

Fondamentalmente, le tecniche utilizzate in quegli anni dall'intelligenza artificiale si basavano su algoritmi di ricerca su strutture dati ad albero, funzioni euristiche per la valutazione dei risultati, manipolazione e combinazione di regole predefinite. I problemi di tale approccio sono emersi quando i ricercatori hanno cominciato ad utilizzare quegli stessi algoritmi per programmare dei robot a svolgere attività senso-motorie complesse, come la manipolazione di oggetti, la locomozione, il controllo dell'equilibrio, il riconoscimento di immagini e la comprensione del parlato. Gli stessi algoritmi in grado di sconfiggere il campione del mondo di scacchi fallivano miseramente se applicati al riconoscimento di forme o al controllo motorio. Successivamente si è capito che la ragione di tale fallimento è dovuta al fatto che il numero delle possibili situazioni da considerare nello svolgimento di attività senso-motorie è talmente elevato che non è possibile codificare il comportamento di un sistema mediante un insieme di regole predefinite.

Si consideri, ad esempio, di voler sviluppare un programma per il riconoscimento di caratteri manoscritti. La Figura 1 mostra alcuni esempi di immagini ($28 \times 28$ pixel, ciascuno con 256 livelli di grigio) di caratteri numerici scritti a mano, illustrando la grande differenza che può esistere tra i modi di scrivere lo stesso carattere. Ora immaginiamo di impostare il riconoscimento sulla base di regole del tipo

- **IF** (esiste un tondino in alto a sinistra) **AND** (esiste una linea verticale incurvata verso sinistra) **THEN** (il carattere è il nove).

Affinchè tale regola possa essere compresa da un algoritmo, occorre renderla più precisa, specificando il significato delle frasi "un tondino in alto a sinistra" e "una linea verticale incurvata verso sinistra". Dovrebbe risultare chiaro che, più si cerca di rendere precisa la regola, più sono i casi particolari che occorre considerare per definire il significato delle frasi. Tale approccio è quindi destinato a generare una quantità esorbitante di casi particolari, eccezioni e sottoregole che comunque non coprirebbero tutte le possibilità che si possono presentare. Basti pensare che una piccola matrice binaria di $16 \times 16$ pixel, ciascuno con soli due livelli di grigio (bianco e nero) può rappresentare ben $2^{16\times16} \sim 10^{77}$ immagini diverse, ossia un numero paragonabile al numero di atomi presenti nell'intero universo (stimato tra $10^{79}$ e $10^{81}$). Dunque l'approccio a regole è destinato a fallire su questa tipologia di problemi in cui il numero di casi possibili cresce esponenzialmente con la dimensione del dato.

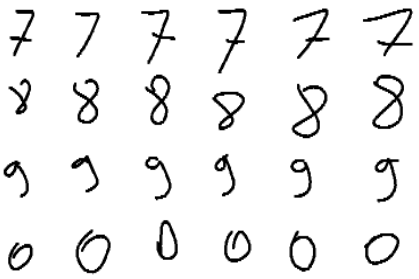

**Figura 1:** *Esempi di caratteri scritti a mano rappresentati da immagini di $28 \times 28$ pixel con 256 livelli di grigio.*

Tuttavia, il cervello umano riesce a risolvere i problemi di riconoscimento e coordinamento senso-motorio in modo rapido ed efficiente. Questa semplice osservazione, unita alle difficoltà di risolvere tali problemi con l'approccio a regole, ha portato i ricercatori a sviluppare dei modelli computazionali ispirati al funzionamento del cervello. Il paragrafo successivo presenta una panoramica storica dei risultati più significativi sulle reti neurali, ottenuti dagli inizi degli anni 40 fino ad oggi.

## Evoluzione della ricerca sulle reti neurali

### Il neurone binario a soglia

Il primo modello di neurone artificiale, noto come neurone binario a soglia, è stato proposto nel 1943 da due ricercatori statunitensi, `Warren McCulloch` (un neurofisiologo) and `Walter Pitts` (un matematico) [1]. Il modello, schematicamente illustrato in Figura 2, consiste in un elemento di calcolo (neurone artificiale) che riceve n valori di ingresso $(x_1, x_2, \ldots, x_n)$ attraverso altrettanti canali che rappresentano i den-

driti di un neurone biologico. Ciascun valore $x_i$ viene modulato da un peso (*weight*) $w_i$, che modella la connessione sinaptica presente sul canale dendritico. Gli $n$ valori di ingresso, opportunamente pesati, vengono poi sommati tra loro per produrre il valore di attivazione $a = \sum_{i=1}^{N} w_i x_i$, equivalente al potenziale di membrana di un neurone biologico. Un neurone biologico produce

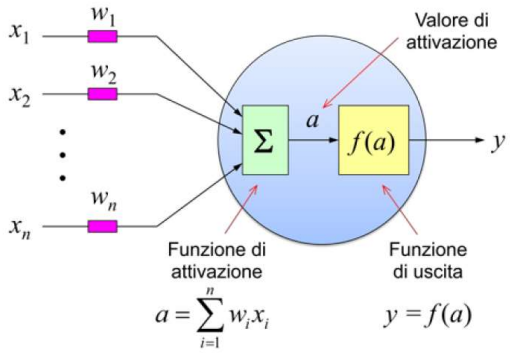


**Figura 2:** *Modello di neurone binario a soglia proposto da McCulloch e Pitts nel 1943.*

un segnale di uscita (*spike*) quando il potenziale di membrana supera un certo livello di soglia. Analogamente, nel modello di McCulloch e Pitts, il valore di uscita $y$ del neurone viene calcolato come $y = f(a)$, dove $f(.)$ è detta funzione di uscita. Nel neurone binario a soglia, come funzione di uscita si utilizza la funzione di Heaviside, illustrata in Figura 3, corrispondente ad un gradino con soglia $\theta$. L'uscita di un neurone binario a

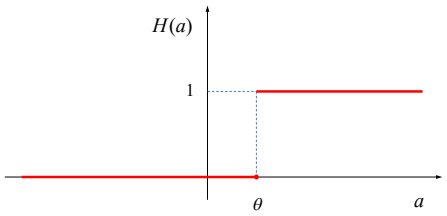


**Figura 3:** *Funzione di Heaviside, utilizzata come funzione di uscita nel neurone binario a soglia.*

soglia, pertanto, può essere espressa come

$$y = +1 \quad \text{Se} \sum_{i=1}^{N} w_i x_i > \theta,$$
$$y = 0 \quad \text{Altrimenti.}$$

È importante osservare che una differenza sostanziale tra un neurone biologico e il modello binario a soglia è che il neurone biologico codifica l'informazione in frequenza, trasmettendo sull'assone una sequenza di *spike* con frequenza proporzionale ai segnali ricevuti in ingresso, mentre il neurone binario a soglia codifica l'informazione in ampiezza, quantizzata su due valori di uscita $(0)$ e $(1)$. Un'altra differenza importante è che il modello di McCullogh e Pitts non è in grado di apprendere, anche perchè in quegli anni non erano stati ancora compresi i meccanismi dell'apprendimento.

## La scoperta di Hebb

Nel 1949, lo psicologo canadese `Donald Hebb` [2] fece una scoperta rivoluzionaria che avanzò le conoscenze sul cervello e fece progredire la ricerca sulle reti neurali. Hebb scoprì che il processo di apprendimento non modifica di fatto il funzionamento delle cellule nervose, ma opera unicamente sulle connessioni sinaptiche che modulano la comunicazione tra i neuroni. Egli riportò che

> "Quando un assone di una cellula A è abbastanza vicino da eccitare una cellula B e partecipa ripetutamente alla sua attivazione, si osservano alcuni processi di crescita o cambiamenti metabolici in una o entrambe le cellule tali da aumentare l'efficacia di A nell'attivare B."

Questa nuova conoscenza sui meccanismi dell'apprendimento ha consentito di integrare questa capacità anche nei modelli neurali allora sviluppati.

## Il Perceptron

Grazie alla scoperta di Hebb, nel 1957, lo psicologo statunitense `Frank Rosenblatt` [3] sviluppò il primo modello di neurone artificiale in grado di apprendere, il **Perceptron**, illustrato in Figura 4. La regola di apprendimento, nota come **Delta Rule** [4], è semplice ma efficace e consiste nel modificare i pesi in modo proporzionale all'ingresso e all'errore $(\delta)$ commesso dal neurone, pari alla differenza tra l'uscita reale $(y)$ e quella desiderata $(y_d)$. La costante di proporzionalità

è detta *learning rate*. In un esperimento che di-

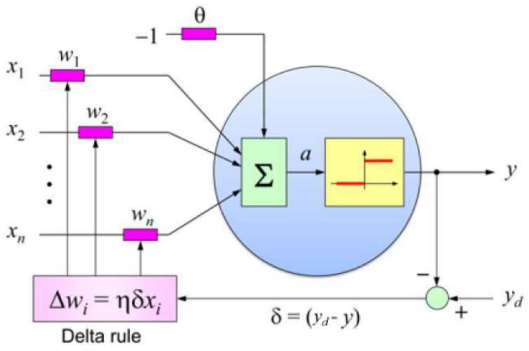


**Figura 4:** *Il Perceptron di Rosenblatt. Durante la fase di apprendimento, i pesi vengono modificati in funzione dell'errore* $(\delta)$ *commesso dal neurone, pari alla differenza tra l'uscita desiderata* $(y_d)$ *e quella reale* $(y)$.

venne poi famoso, Rosenblatt costruì il primo `Perceptron` in *hardware*, realizzando i pesi con dei potenziometri motorizzati e collegando gli ingressi a 400 fotocellule disposte come una matrice di $20 \times 20$ *pixel*. Presentando in ingresso i disegni di varie forme geometriche, Rosenblatt riuscì ad addestrare il `Perceptron` a riconoscere le forme concave da quelle convesse[1].

## Il controesempio di Minsky e Papert

Gli entusiasmi di Rosenblatt purtroppo svanirono nel 1969, quando due matematici del Massachusetts Institute of Technology (MIT), `Marvin Minsky` e `Seymour Papert`, pubblicarono un libro intitolato "Perceptrons" [5], in cui venivano dimostrati formalmente i punti di forza ma anche i maggiori limiti del modello di Rosenblatt. In particolare, Minsky e Papert dimostrarono, attraverso un controesempio, l'impossibilità per un `Perceptron` di apprendere la semplice funzionalità di un OR esclusivo (XOR) a due ingressi, che prevede una risposta pari a zero quando i due ingressi sono uguali (entrambi zero o entrambi uno) e una risposta pari a uno quando i due ingressi sono diversi.

Questo risultato negativo sul Perceptron fece precipitare l'interesse per le reti neurali per oltre un decennio, dal 1969 al 1982, periodo che oggi viene indicato come *AI winter*.

## Le reti di Hopfield

L'interesse per le reti neurali si riaccese nel 1982, quando `John Hopfield` [6] propose un nuovo modello di rete in grado di comportarsi come una memoria associativa, ossia una memoria in cui è possibile memorizzare un insieme di informazioni desiderate per poi recuperarle partendo da dati parziali o distorti. Se ad esempio in una rete di Hopfield vengono memorizzate delle immagini, queste possono poi essere recuperate fornendo in ingresso alla rete delle immagini simili, rumorose o distorte.

Hopfield dimostrò che tale proprietà può essere ottenuta costruendo una rete di neuroni binari a soglia che soddisfi le seguenti proprietà:

1. tutti i neuroni sono connessi tra loro;
2. la funzione di uscita è la funzione segno;
3. ogni coppia di neuroni ha pesi simmetrici;
4. i neuroni cambiano stato uno per volta.

Se queste proprietà vengono rispettate, è possibile dimostrare che, partendo da un qualsiasi stato iniziale, la rete evolve attraverso una serie di commutazioni, generando una sequenza di stati (traiettoria) che termina sempre in uno stato stabile.

Hopfield fornì anche una regola per poter rendere stabili degli stati neurali desiderati, che rappresentano quindi le memorie della rete. Per rendere stabile una configurazione di attivazioni neurali, egli suggerì di collegare neuroni con attivazione simile con pesi positivi e neuroni con attivazione opposta con pesi negativi. Per memorizzare più informazioni stabili basterà sommare i pesi ottenuti per ciascuno stato.

La Figura 5 illustra un esempio di come la rete di Hopfield sia in grado di recuperare l'immagine del numero tre (precedentemente memorizzata come stato stabile) partendo da un'immagine notevolmente rumorosa presentata in ingresso.

## Le reti di Kohonen

Sempre nel 1982, `Teuvo Kohonen` propose un modello di rete neurale [7] capace di autoorganizzarsi per formare delle mappe sensoriali

[1] Si ricorda che una forma si dice convessa se, presi due punti A e B al suo interno, il segmento che li unisce è contenuto tutto all'interno della figura. Viceversa la figura si dice concava.

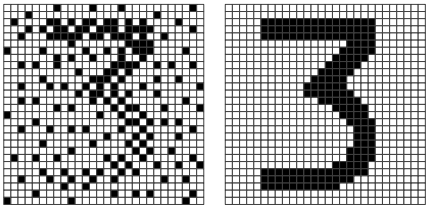

**Figura 5:** *Esempio di memoria associativa realizzata mediante una rete di Hopfield: (a) immagine presentata come stato iniziale; (b) immagine recuperata, corrispondente ad uno stato stabile memorizzato in precedenza. Ogni pixel corrisponde ad un neurone. In questo esempio la rete ha 784 neuroni per rappresentare immagini binarie di* $28 \times 28$ *pixel.*

simili a quelle esistenti nella corteccia somatosensoriale, sulla quale viene rappresentato il cosiddetto homunculus sensitivo. Una rete neurale di Kohonen è formata da due soli strati: uno stato di ingresso e uno di uscita, come rappresentato in Figura 6. La regola di apprendimento è tale

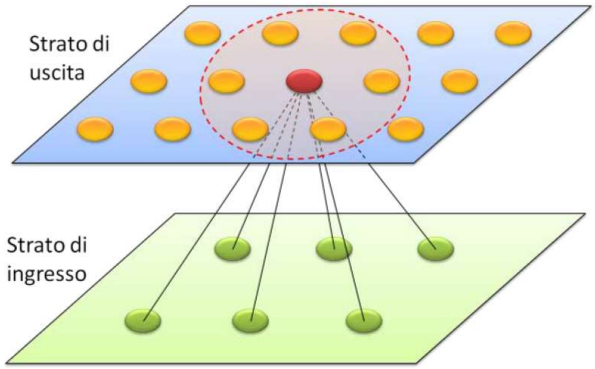


**Figura 6:** *Esempio di una rete di Kohonen con 6 neuroni di ingresso e 15 neuroni di uscita disposti su una mappa bidimensionale.*

da creare un isomorfismo tra stimoli sensoriali di ingresso e neuroni di uscita, per cui neuroni vicini si specializzano a riconoscere stimoli sensoriali simili. Questa proprietà viene ottenuta attraverso un meccanismo di apprendimento competitivo che, per ogni stimolo, aggiudica come vincitore il neurone che ha l'attivazione più alta tra tutti. Per ottenere l'isomorfismo, i pesi del neurone vincitore e quelli dei neuroni appartenenti ad un vicinato (illustrato in figura con una linea tratteggiata rossa) vengono modificati in modo che tali neuroni si specializzino ancor meglio a riconoscere quello stimolo.

Le reti di Kohonen assumono una grande rilevanza nel panorama dei modelli neurali, in quanto consentono l'estrazione di caratteristiche salienti dai dati di ingresso senza alcuna supervisione da parte dell'utente. Esse vengono utilizzate per ottenere una compressione dei dati, o un raggruppamento di dati omogenei in un insieme di classi (*clustering*) in base alla somiglianza tra i dati. Esse possono persino essere utilizzate per risolvere efficientemente problemi di ottimizzazione combinatoria.

## Reinforcement Learning

Nel 1983, Andrew Barto, Richard Sutton e Charles Anderson [8] proposero un nuovo modello di rete neurale in grado di generare azioni di controllo utilizzando un paradigma di apprendimento basato su premi e punizioni, e denominato Reinforcement Learning.

L'idea alla base di questo meccanismo è che la rete neurale generi inizialmente delle azioni casuali di controllo e riceva una ricompensa (un segnale di *feedback* positivo) o una punizione (un segnale di *feedback* negativo) in base all'esito di tali azioni. I segnali di *feedback* ricevuti vengono utilizzati per modificare i pesi della rete in modo da favorire le azioni che hanno generato una ricompensa e scoraggiare quelle che hanno generato una punizione. In questo modo la rete si costruisce gradualmente una conoscenza del sistema, passando da una fase esplorativa, pesantemente guidata dal caso, ad una fase operativa, in cui la conoscenza acquisita viene sfruttata per generare le azioni migliori.

Dal 1983 ad oggi, questo paradigma di apprendimento si è notevolmente evoluto, grazie anche alle tecniche più avanzate di *deep learning* sviluppate di recente, raggiungendo prestazioni eccellenti in diverse applicazioni. In particolare, nel 2010, la DeepMind Technology ha utilizzato questa metodologia per addestrare una rete neurale a giocare a numerosi videogiochi Atari, riuscendo a superare le prestazioni umane in ben sette di essi. Nel 2014, l'azienda è stata acquisita da Google e il 23 maggio 2017 la Google DeepMind è ritornata alla ribalta per aver costruito una rete basata su *reinforcement learning*, denominata

AlphaGo Zero, in grado di battere il campione del mondo di Go, Ke Jie. Questo risultato è alquanto rilevante, poichè il Go è uno tra i giochi più complessi al mondo, in cui l'albero delle possibili posizioni è dell'ordine di $10^{170}$, contro il $10^{120}$ degli scacchi.

Oggi, il *reinforcement learning* è tra gli algoritmi maggiormente studiati, in quanto promette di risolvere una grande quantità di problemi rilevanti e difficili, tra cui la l'apprendimento della camminata in robot bipedi, la guida di veicoli autonomi e il controllo di sonde esplorative spaziali, solo per citarne alcuni.

## La Backpropagation

Nel 1986, David Rumelhart, Geoffrey Hinton e Ronald Williams [9] svilupparono un potente algoritmo di apprendimento supervisionato, noto come Backpropagation, che permette ad una rete neurale di imparare a classificare dei pattern di ingresso attraverso un insieme di esempi, detto *training set*.

Le reti neurali addestrabili con Backpropagation sono di tipo stratificato, come quella illustrata in Figura 7. Ogni neurone di uno strato è connesso con ogni neurone dello strato successivo, ma non esistono connessioni tra neuroni dello stesso strato, nè tra neuroni appartenenti a strati non adiacenti. Il primo strato è quello di ingresso *(input layer)*, che riceve i dati da elaborare. L'ultimo strato è quello di uscita *(output layer)*, che produce i risultati dell'elaborazione. Gli strati intermedi vengono detti strati nascosti *(hidden layer)* in quanto non sono visibili dall'esterno in una visione *black-box* della rete. In questo tipo di rete, il modello di neurone utilizzato in tutti gli strati è molto simile al neurone binario a soglia e differisce unicamente per la funzione di uscita. Le funzioni di uscita oggi più utilizzate sono la sigmoide, la tangente iperbolica e la lineare rectificata (ReLU), illustrate in Figura 8. L'aspetto più interessante dell'apprendimento supervisionato è che la rete riesce a generalizzare ciò che ha appreso, classificando correttamente nuovi dati mai visti in fase di *training*.

Grazie a questi risultati, nei vent'anni successivi alla nascita della Backpropagation, le reti neurali sono state utilizzate per risolvere diverse

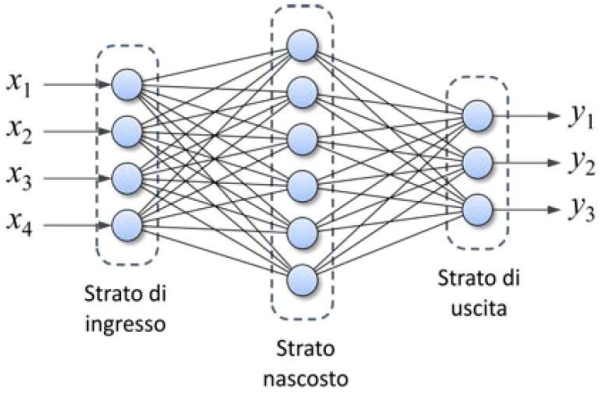


**Figura 7:** *Esempio di rete a tre strati addestrabile con l'algoritmo di Backpropagation.*

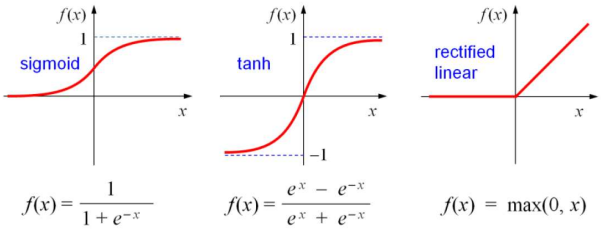


**Figura 8:** *Funzioni di uscita comunemente utilizzate nelle reti multistrato addestrate con Backpropagation.*

tipologie di problemi, tra cui il riconoscimento di immagini, la compressione di dati, la previsione di segnali e serie storiche e il controllo di sistemi robotici, nei più disparati settori, quali fisica, chimica, ingegneria, geologia, agraria, astronomia, economia, medicina, scienze sociali, psicologia, ecc.

## Nuovi ostacoli

Nonostante l'esplosione dei campi applicativi, dal 1986 al 2006 non ci furono nuovi sostanziali sviluppi teorici sulle reti neurali. Molti ricercatori provarono a sviluppare modelli neurali più complessi, più vicini alla controparte biologica, ma non si riusciva ad ottenere dei comportamenti interessanti e, soprattutto, a dimostrare delle proprietà generali sulle le quali costruire algoritmi generalizzabili. Altri ricercatori provarono ad aumentare il numero di strati di una rete addestrata con l'algoritmo di Backpropagation, ma osservarono grosse difficoltà ad addestrare reti con più di quattro strati. Tali problemi sono stati risolti solo negli anni 2000.

## La rivoluzione delle deep neural network

A partire dal 2006, la ricerca sulle reti neurali ha avuto una grossa impennata grazie e tre importanti fattori:

1. Compresi i problemi che causavano la difficoltà di addestrare reti con più di quattro strati, sono state sviluppate nuove metodologie in grado di superare quei limiti e gestire l'apprendimento di reti molto più grandi, costituite da migliaia di neuroni organizzati su numerosi strati: le `deep neural network`.

2. Le `deep neural network`, essendo costituite da migliaia di neuroni, richiedono una grossa potenza di calcolo per essere addestrate. Intorno al 2006, la potenza di calcolo necessaria è diventata disponibile a basso costo grazie alla produzione di nuove piattaforme di calcolo basate sulle `Graphics Processing Unit` (GPU). Tali piattaforme, originariamente progettate per gestire operazioni grafiche, sono state modificate per poter svolgere anche calcoli vettoriali, come quelli necessari in una rete neurale.

3. I primi risultati ottenuti con le `deep neural network` hanno attratto l'interesse di grosse aziende, come `Google`, `Microsoft` e `Facebook` che, gestendo un'enorme quantità di dati, hanno visto nelle reti neurali una grossa opportunità per risolvere problemi di classificazione di immagini, riconoscimento di volti, suoni, voci, e hanno quindi cominciato ad investire grosse quantità di risorse in questo settore.

Infine, un altro elemento che ha contribuito all'evoluzione delle `deep network` è stata la competizione internazionale `ImageNet`, o più precisamente la `ImageNet Large-Scale Visual Recognition Challenge` (ILSVRC) [10], una sorta di olimpiade annuale della *computer vision*, nata nel 2010 per mettere in competizione i migliori gruppi al mondo su problemi complessi relativi al riconoscimento di immagini. Ogni squadra aveva a diposizione un enorme database e doveva addestrare la propria rete su un milione di immagini suddivise in mille categorie. Una volta addestrate, le reti dovevano classificare 100.000 nuove immagini. Poichè un'immagine poteva contenere diversi oggetti, la risposta della rete era considerata corretta se la classe veniva correttamente identificata considerando le 5 uscite con valore più elevato (sulle mille possibili). La Figura 9 illustra come si è ridotto l'errore di classificazione delle reti vincitrici dal 2010 al 2017.

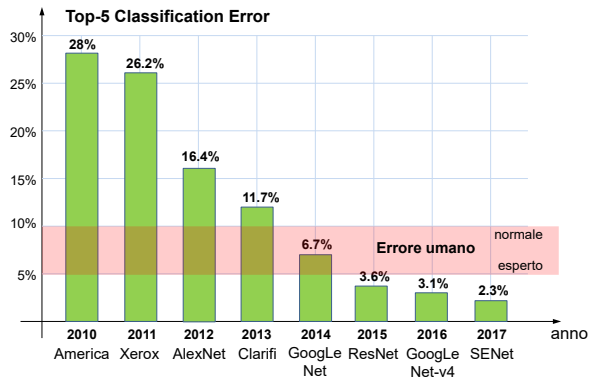


**Figura 9:** *Diminuzione dell'errore di classificazione delle reti neurali dal 2010 al 2017 nella competizione ImageNet.*

È interessante notare come, per la prima volta nella storia, nel 2014 la rete `GoogLeNet` ha eguagliato le prestazioni umane, poi superate negli anni successivi da altre reti.

# Paradigmi di Apprendimento

Negli anni sono stati proposti diverse modalità di apprendimento, che possono essere ricondotte e tre paradigmi principali:

1. apprendimento con supervisione;

2. apprendimento senza supervisione;

3. apprendimento con rinforzo.

I concetti alla base di questi meccanismi sono descritti di seguito.

## Apprendimento supervisionato

L'obbiettivo dell'apprendimento supervisionato è quello di addestrare una rete neurale a riconoscere dei dati di ingresso come appartenenti a delle `categorie (o classi)` predefinite. Tali categorie vengono mostrate alla rete mediante un insieme di esempi, detto *training set*.

Il funzionamento della rete è suddiviso in due fasi, dette `di addestramento` e `di inferenza`. Nella fase di addestramento vengono presentati gli esempi del training set dai quali la rete deve imparare. Ciascun esempio consiste in una coppia di dati vettoriali: l'ingresso da classificare $(x)$ e l'uscita desiderata da associare $(y_d)$. Per ogni esempio viene calcolata l'uscita della rete $(y)$ e tali valori sono utilizzati per calcolare una `funzione di errore` (detta anche `loss function`) con cui modificare i pesi della rete.

Detta $E$ la funzione che descrive l'errore della rete rispetto all'uscita desiderata $y_d$, ciascun peso viene modificato in modo da diminuire l'errore, calcolando il gradiente della funzione $E$ rispetto al peso. Uno schema del paradigma supervisionato è illustrato in Figura 10. Ad ogni iterazione l'errore tende a diminuire e, quando esso diventa inferiore ad una certa soglia prestabilita, la rete si considera addestrata e può essere utilizzata per la fase di inferenza. In questa fase, la rete viene utilizzata per effettuare la classificazione vera e propria di nuovi dati. Al fine di comprendere

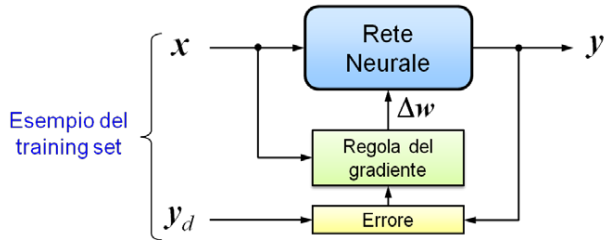


**Figura 10:** *Paradigma di apprendimento supervisionato.*

meglio il paradigma supervisionato, si consideri una rete con due ingressi $(x_1, x_2)$ e una sola uscita $(y)$. Tale rete può essere addestrata a riconoscere se un certo insieme di dati appartiene ad una certa classe A $(y_d = 1)$ oppure no $(y_d = 0)$. Visto che ogni dato d'ingresso è descritto da una coppia di coordinate, gli esempi possono essere rappresentati su un piano cartesiano con dei punti: se un dato appartiene alla classe A allora il punto viene indicato con il colore rosso, altrimenti con il colore nero. La situazione descritta è illustrata in Figura 11.

Terminato l'apprendimento, nella fase di inferenza la rete deve stabilire se dei nuovi dati (rappresentati in figura con dei punti grigi) appartengano o meno alla classe A.

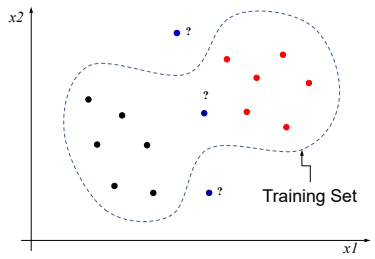


**Figura 11:** *Problema della classificazione nel paradigma di apprendimento supervisionato. I punti colorati rappresentano gli esempi del training set descritti da due variabili $(x_1, x_2)$: i punti rossi appartengono ad una classe A, i punti neri no. Terminato l'apprendimento, nella fase di inferenza la rete deve stabilire se dei nuovi dati (i punti blu) appartengono o meno alla classe A.*

## Apprendimento senza supervisione

Nell'apprendimento senza supervisione gli esempi utilizzati per addestrare la rete non sono etichettati come appartenenti ad una classe, per cui non esiste un'uscita desiderata per ogni dato di ingresso. In questo caso, la rete deve imparare a suddividere i dati in diversi gruppi (*cluster*) sulla base della loro somiglianza. La Figura 12 illustra un caso in cui i 12 esempi forniti alla rete vengo raggruppati in due *cluster* di 6 elementi ciascuno. La figura mostra anche come,

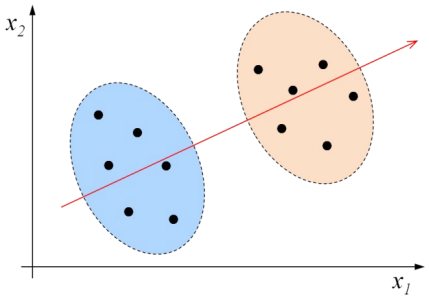


**Figura 12:** *Problema del* clustering *nel paradigma di apprendimento non supervisionato. In questo caso i dati non sono etichettati come appartenenti ad una classe. Lo scopo della rete è di suddividerli in gruppi (cluster) sulla base della loro somiglianza.*

in molti casi, la rete può rilevare una ridondanza nei dati e operare una riduzione di dimensioni. Nel caso di figura, ad esempio, i *cluster* possono essere separati utilizzando una sola variabile

(rappresentata dall'asse indicato in rosso) ottenuta come combinazione lineare delle due variabili originali. Questa proprietà, comune a molte reti non supervisionate, rende questo meccanismo di apprendimento particolarmente adatto alla compressione di dati oppure all'estrazione di caratteristiche salienti dai dati di ingresso.

### Apprendimento con Rinforzo

Il paradigma di apprendimento con rinforzo si utilizza tipicamente nei problemi di controllo, ossia quando vogliamo addestrare una rete neurale ad inviare delle azioni di comando ad un sistema che interagisce con un ambiente. Tale modalità di apprendimento è una via di mezzo fra le due precedenti, poichè richiede solo una leggera supervisione, che però non necessita di fornire la risposta desiderata per ognuno degli esempi del training set. Per ogni azione generata sul sistema, la rete riceve una valutazione da parte di un `critico`, la cui funzione è solo quella di accorgersi quando il sistema fallisce oppure raggiunge un obiettivo. Tale valutazione è codificata in un `segnale di rinforzo` o *reward* che viene utilizzato per modificare i pesi della rete.

Per fare un esempio concreto, si consideri lo schema illustrato in Figura 13, supponendo di utilizzare l'uscita della rete neurale per controllare lo sterzo di un auto. Al fine di poter imparare a sterzare correttamente, la rete dovrà ricevere delle informazioni sullo stato dell'auto, ad esempio le immagini prelevate da una telecamera che inquadra la strada. In questo caso, il segnale di rinforzo $(R)$ prodotto dal critico potrebbe essere negativo $(-1)$ quando l'auto esce fuori strada e positivo $(1)$ quando l'auto riesce a tenersi al centro della carreggiata. In tutti gli altri casi, il segnale di rinforzo può essere nullo. L'obiettivo dell'apprendimento con rinforzo è quindi quello di imparare a generare azioni che migliorino la valutazione del critico nel tempo. L'esempio illustrato suggerisce come questa modalità di apprendimento sia paragonabile a quella basata su premi e punizioni. Un rinforzo positivo ricevuto dal critico è assimilabile ad un premio, mentre un rinforzo negativo è assimilabile ad una punizione. Il meccanismo di apprendimento è tale da scoraggiare la rete a ripetere le azioni che in certo stato hanno generato fallimenti, favorendo invece le azioni che hanno causato delle valutazioni positive.

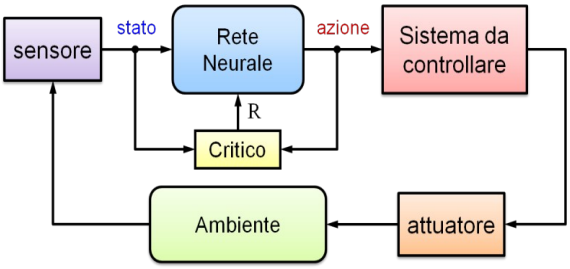


**Figura 13:** *Paradigma di apprendimento con rinforzo. La rete neurale genera delle azioni di controllo su un sistema che interagisce con l'ambiente e riceve da un critico una valutazione $(R)$ sulla loro efficacia. Lo scopo dell'apprendimento con rinforzo è di imparare a generare azioni che migliorino la valutazione del critico.*

Considerato che il critico può essere facilmente realizzato elaborando i dati prodotti da opportuni sensori (ad esempio sensori di contatto, accelerazione, distanza, ecc.) questo paradigma di apprendimento risulta molto potente, poichè in grado di scoprire le azioni corrette senza l'intervento umano, ma unicamente sulla base dei fallimenti e dei successi sperimentati dal sistema.

## I modelli recenti che hanno rivoluzionato l'intelligenza artificiale

Dal 2000 ad oggi sono stati ideati nuovi modelli di rete neurale che hanno permesso di risolvere problemi prima considerati intrattabili con queste tecniche. In ordine temporale, i modelli più rilevanti proposti in letteratura sono le reti ricorrenti, le reti convoluzionali e le reti generative.

### Le Reti Ricorrenti

A differenza delle reti neurali considerate finora in questo articolo, le `reti neurali ricorrenti`, o *Recurrent Neural Networks* (RNN) sono in grado di trattare sequenze temporali di dati sia in

ingresso che in uscita. Per ottenere questo scopo, le reti ricorrenti posseggono delle connessioni aggiuntive, rispetto alle reti multistrato, che collegano le uscite dei neuroni nascosti ai neuroni di ingresso. La Figura 14 illustra la struttura di una rete ricorrente, in cui l'uscita dello strato nascosto $\mathbf{h}(t)$ viene riportata in ingresso $(a)$. Una rete ricorrente viene spesso rappresentata in una modalità sviluppata nel tempo (*unrolled*), in cui sono evidenziati i valori dei vettori per i vari istanti temporali $(b)$. Tra i moti convoluzionali, strati di subsampling e strati completamente connessi.

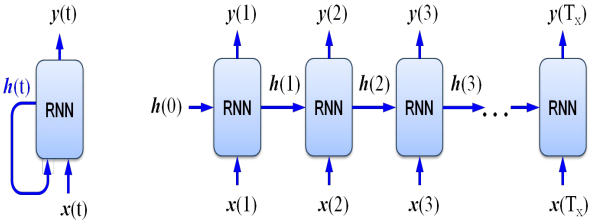


**Figura 14:** *Struttura di una rete ricorrente in cui l'uscita dello strato nascosto* $\mathbf{h}(t)$ *viene riportata in ingresso* $(a)$*. La figura* $(b)$ *illustra la versione* unrolled *in cui sono evidenziati i valori dei vettori per i vari istanti temporali.*

delli più utilizzati per le reti ricorrenti ricordiamo le `Long short-term memory` (LSTM) [11] e le `Gated Recurrent Units` (GRU) [12]. Tra le applicazioni più interessanti di questi modelli citiamo l'analisi e la previsione di testi, il riconoscimento vocale, la traduzione automatica, il riconoscimento di sequenze video, la descrizione testuale di immagini e la composizione musicale.

## Le Reti Convoluzionali

Le reti convoluzionali sono state introdotte nel 1998 da `Yann LeCun` et al. [13], sebbene delle versioni preliminari fossero state già utilizzate nel 1970. Tali reti sono particolarmente indicate per la classificazione di immagini, in quanto fanno uso di una speciale architettura che sfrutta la struttura spaziale delle immagini per ridurre il numero di connessioni tra neuroni. La tipica architettura di una rete convoluzionale consiste in una sequenza di strati di vario tipo, tra cui strati convoluzionali, strati di subsampling e strati completamente connessi.

Gli strati convoluzionali effettuano un'estrazione di caratteristiche dallo strato precedente attraverso un'operazione di convoluzione, che consiste nel moltiplicare i valori $x$ di una piccola area di dimensione $r \times r$ (*receptive field*) per una matrice $w$ di pesi (*weights*) della stessa dimensione, detta *kernel* o `filtro`. Tale filtro viene traslato sullo strato in modo da estrarre la stessa caratteristica in zone diverse dello strato. In questo modo, il numero di pesi da modificare risulta pari ad $r^2$ ed è indipendente dalle dimensioni dello strato.

La Figura 15 illustra un esempio di convoluzione su uno strato di $129 \times 129$ neuroni, utilizzando un filtro di dimensioni $3 \times 3$ traslato di 2 neuroni per volta. Lo strato convoluzionale risulta pertanto composto da $64 \times 64$ neuroni, che costituiscono una mappa di caratteristiche (*feature map*) rilevate in diverse posizioni spaziali. Ciascun neurone dello strato convoluzionale risulta quindi un rivelatore della caratteristica codificata nei pesi nel filtro: un valore elevato indica che quella caratterista è presente nel campo recettivo corrispondente. Gli strati di *subsampling* effettuano

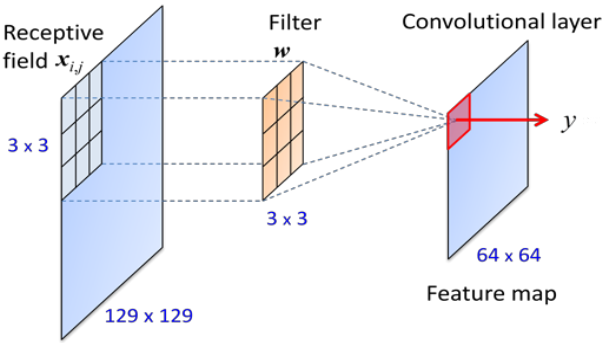


**Figura 15:** *Operazione di convoluzione elementare in una rete convoluzionale. Il valore* $y$ *del neurone in rosso è calcolato come la somma dei prodotti tra i valori dei neuroni del campo recettivo* $x$ *e i pesi del filtro* $w$*, i.e.* $y = \mathbf{x_{i,j}} * \mathbf{w} = \sum_{l=1}^{r} \sum_{m=1}^{r} x_{i+l,j+m} w_{lm}$.

una compressione dell'informazione, riducendo il numero di neuroni dello strato precedente attraverso operazioni di media o di massimo su piccole aree neuronali. Una rete di solito utilizza diversi strati convoluzionali seguiti da altrettanti strati di *subsampling*. Gli strati completamente connessi realizzano infine la classificazione vera

e propria e sono utilizzati nella parte finale della rete. La rete termina con lo strato di uscita, il cui numero di neuroni è pari al numero di categorie che si desidera riconoscere nelle immagini.

Un'altra funzione importante realizzabile attraverso le reti convoluzionali è quella di rilevare anche la posizione dell'oggetto riconosciuto (*object detection*) attraverso un rettangolo (*bounding box*) descritto da quattro coordinate, due per il centro e due per le dimensioni dei lati. In questa modalità operativa, se C è il numero di categorie da riconoscere, lo strato di uscita dovrà contenere almeno $C + 4$ neuroni.

Oggi le reti convoluzionali hanno raggiunto prestazioni eccellenti in diversi settori applicativi, tra cui quello medico, in cui le reti neurali sono state utilizzate per effettuare diagnosi precoci a partire da immagini mediche e scansioni tomografiche. In particolare sono state ottenute prestazioni paragonabili o superiori a quelle umane nell'analisi di elettrocardiogrammi, nell'identificazione di tumori della pelle e di patologie della retina, del cancro al polmone e nella diagnosi dell'Alzheimer.

## Le Reti Generative

Le reti generative o *Generative Adversarial Networks* (GAN) sono una particolare classe di reti neurali ideate nel 2014 da `Ian Goodfellow` et al. [14] al fine di produrre nuovi campioni di dati aventi la stessa distribuzione statistica di quelli utilizzati per l'addestramento. In altre parole, se una GAN viene addestrata utilizzando un database di volti umani, alla fine dell'addestramento essa sarà in grado generare delle nuove immagini realistiche di volti umani; se addestrata con foto di paesaggi naturali, essa potrà generare foto di nuovi paesaggi.

Ma l'utilizzo delle GAN non si riduce a questo. Negli ultimi anni esse sono state utilizzate nelle più svariate applicazioni, tra cui la colorazione di immagini e filmati in bianco e nero (*colorization*), la generazione di immagini a risoluzione più elevata di quella del campione in ingresso (*super resolution*), il restauro di foto danneggiate (*pixel restoration*), la generazione di voci e musica, l'animazione di volti dipinti (*reenactment*), o la trasformazione di foto in quadri stile Van Gogh o Monet (*style transfer*).

Una rete GAN è composta in realtà da due tipi di reti neurali, un `Generatore` ed un `Discriminatore`, che competono tra loro in una sorta di gioco. Il `Generatore` gioca il ruolo di un falsario che produce delle opere false che vuole far passare come autentiche, mentre il `Discriminatore` agisce come un ispettore che cerca di identificare le opere false. La regola di apprendimento è strutturata in modo che entrambe le reti siano portate a migliorare le loro capacità, fino al punto che le opere false diventano indistinguibili da quelle autentiche.

Il `Discriminatore` è addestrato in modo supervisionato per distinguere i campioni veri dai falsi. Esso ha quindi un solo neurone di uscita, che vale 1 quando il campione d'ingresso è autentico e 0 quando è falso. Il `Generatore` è invece addestrato con una modalità non supervisionata, modificando i pesi in modo da favorire la generazione di quei campioni che hanno ingannato il `Discriminatore` e penalizzare la generazione di quelli che sono stati intercettati come falsi. Una volta che la GAN è stata addestrata, il `Discriminatore` viene eliminato, in quanto lo scopo finale è quello di generare campioni realistici. Lo schema architetturale di una GAN è illustrato sinteticamente in Figura 16.

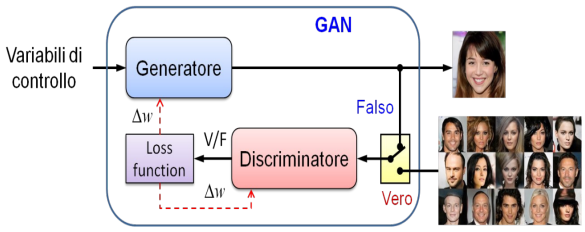


**Figura 16:** *Architettura di una rete GAN. Il Generatore ha il compito di produrre campioni falsi realistici, mentre il Discriminatore ha il compito di distinguere i campioni autentici (provenienti dal database) da quelli falsi prodotti dal Generatore. Gli errori commessi da ciascuna rete contribuiscono al miglioramento di entrambe.*

# Problemi irrisolti

Nei paragrafi precedenti sono state descritte le capacità di elaborazione delle reti neurali in nu-

merosi campi applicativi. Grazie a queste potenzialità, l'industria sta considerando seriamente di utilizzare questa metodologia per sviluppare robot intelligenti e veicoli a guida autonoma. Tuttavia, quando si tratta di realizzare sistemi che devono interagire con l'uomo, occorre garantire non solo prestazioni elevate, ma soprattutto altre proprietà particolarmente critiche, quali predicibilità, affidabilità, e sicurezza.

Fino ad oggi le reti neurali e la maggior parte degli algoritmi di intelligenza artificiale sono stati utilizzati per realizzare applicazioni non critiche, come il riconoscimento di caratteri manoscritti, il riconoscimento facciale, la traduzione automatica, e il riconoscimento vocale. Un errore commesso da una rete in una di queste applicazioni non comporta danni per l'uomo. Immaginiamo invece cosa potrebbe accadere se una rete dovesse commettere un errore di riconoscimento o di controllo in un'automobile a guida autonoma. Del resto, gli incidenti che si sono verificati di recente su automobili avanzate, come la Tesla, che consentono di attivare funzionalità automatiche (sotto la stretta supervisione del conducente), indicano che questi sistemi possono fallire in casi particolari (indicati come *corner case*) in cui più condizioni non previste contribuiscono a produrre un malfunzionamento.

Si consideri, ad esempio, l'incidente verificatosi il 7 Maggio 2016, in cui Joshua Brown è stato vittima di un incidente mortale mentre era alla guida della sua `Tesla S` a guida autonoma nei pressi di Williston, Florida (USA). In base alla ricostruzione dell'incidente [15] si è capito che un TIR che viaggiava sulla corsia opposta ha attraversato la doppia linea gialla per svoltare a sinistra. Il sistema di visione non lo ha rilevato poichè il cielo era troppo luminoso e il camion era bianco, per cui l'auto non ha frenato e si è scontrata contro il TIR, causando la morte di Brown. In base alle direttive `Tesla`, il Sig. Brown sarebbe dovuto intervenire sui comandi, ma purtroppo non lo ha fatto in quanto era intento a guardare un film.

Al fine di evitare questo tipo di incidenti, le auto del futuro dovranno essere progettate per essere tolleranti ai malfunzionamenti di un singolo componente, prevedendo una ridondanza di sottosistemi basati su tecnologie differenti.

Un altro aspetto essenziale da garantire nei sistemi critici è quello della sicurezza. Nel 2015, due ricercatori di *cyber-security*, Charlie Miller and Chris Valasek, sono riusciti da remoto a compromettere il *software* di controllo di una Jeep Cherokee [17], portandola fuori strada dopo aver assunto il controllo del volante e disabilitato la trasmissione e i freni della vettura. Gli strumenti *software* con i quali oggi vengono gestite le reti neurali non sono progettati per essere sicuri, anzi contribuiscono ad aumentare notevolmente le superfici di attacco al sistema di controllo. Pertanto, occorre investire in questo settore di ricerca per mettere in sicurezza tutti i sistemi basati su intelligenza artificiale e ridurre le probabilità di attacco.

Infine, un altro aspetto relativo alla sicurezza delle reti neurali è legato ad una tecnica in grado di generare immagini (dette *adversarial sample*) [16, 18] che appaiono normali alla vista umana, ma vengono interpretate erroneamente dalla rete neurale, che riconosce l'immagine come appartenente ad una categoria diversa, definita arbitrariamente. Sfruttando questa tecnica, un *hacker* potrebbe decidere di attaccare un veicolo autonomo basato su reti neurali senza intervenire sul *software* di controllo, ma semplicemente agendo sull'ambiente. Basterebbe generare un'immagine avversaria di un segnale stradale, ad esempio lo stop, e attaccarla sul vero segnale, in modo che esso venga interpretato come un albero o un uccello.

Molte delle tecniche utilizzate per generare immagini avversarie si basano sulla conoscenza della rete e dei pesi. Esse sfruttano l'algoritmo di `backpropagation` non per modificare il valore dei pesi della rete, ma per modificare il valore di alcuni `pixel` di un'immagine di rifermento posta in ingresso. I `pixel` vengono modificati per ridurre l'errore su una classe erronea desiderata e aumentarlo sulla quella corretta. L'immagine così modificata risulta pressochè identica a quella di partenza ad un osservatore umano, ma la rete neurale la interpreta in modo totalmente diverso.

I ricercatori hanno già cominciato a studiare nuove tecniche per affrontare questo nuovo tipo di attacco, ma non esiste ancora una soluzione definitiva al problema.

## Conclusioni

In questo articolo è stata presentata una panoramica della ricerca sulle reti neurali, dai primi modelli sviluppati verso la metà del secolo scorso fino alle metodologie più recenti relative alle deep network. In numerosi settori applicativi le reti neurali hanno raggiunto o superato le prestazioni umane e promettono di diventare uno strumento essenziale per la previsione di eventi, le diagnosi mediche, la progettazione di nuovi farmaci, la guida autonoma e la percezione artificiale nei robot del futuro.

Tuttavia, al fine di poter essere utilizzate in sistemi ad elevata criticità, in cui è prevista una stretta interazione con l'uomo, è necessario affrontare nuovi problemi, finora trascurati, quali la predicibilità temporale delle risposte, l'affidabilità di funzionamento e gli aspetti legati alla sicurezza, al momento ancora irrisolti.

Superate queste difficoltà, in un futuro non tanto lontano, dovremo affrontare un problema ancora più grande, che coinvolgerà aspetti legali, sociali, etici e psicologici: la convivenza con entità artificiali, fisiche e virtuali, dotate di intelligenza superiore a quella umana.


[1] W. S. McCulloch and W. Pitts: "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics* **5** (1943) 115.

[2] D. O. Hebb: *The organization of behavior.* Springer, Berlin (1949).

[3] F. Rosenblatt: *The Perceptron – a perceiving and recognizing automaton*, Report 85–460–1, Cornell Aeronautical Laboratory(1957).

[4] F. Rosenblatt: *Principles of neurodynamics.* Spartan, New York (1962).

[5] M. Minsky and S. Papert: *Perceptrons.* MIT press, Cambridge, MA (1969).

[6] J. J. Hopfield: "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences, USA* **79** (1982) 2554.

[7] T. Kohonen: *Self-Organization and Associative Memory.* Springer-Verlag, Berlin (1984).

[8] A. G. Barto, R. Sutton, and W. Anderson: "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Transactions on Systems, Man and Cybernetics* **13** (1983) 834.

[9] Rumelhart D. E., Hinton G. E., and Williams R. J.: "Learning representations by back-propagating errors", *Nature* **323** (1986) 533.

[10] URL: http://www.image-net.org/

[11] S. Hochreiter and J. Schmidhuber: "Long short-term memory", *Neural Computation* **9** (1997) 1735.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio: *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, arXiv:1406.1078v3 (2014).

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: "Gradient-based learning applied to document recognition", *Proc. of the IEEE* **86** (1998) 2278-2324.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio: *Generative Adversarial Nets*, Proceedings of Neural Information Processing Systems, (2014) 8.

[15] The report of the investigation for the Tesla S accident occurred on May 7, 2016 in Florida. U.S. Department of Transportation, National Highway Traffic Safety Administration (NHSA), URL: https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF

[16] C. Szegedy et al.: *Intriguing properties of neural networks*, arXiv:1312.6199v4 (2014).

[17] Black Hat USA 2015 *The full story of how that Jeep was hacked.* `https://www.kaspersky.com/blog/blackhat-jeep-cherokee-hack-explained/9493/`

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy: *Explaining and Harnessing Adversarial Examples*, arXiv:1412.6572v3 (2015).



**Giorgio Buttazzo:** è professore ordinario di Ingegneria Informatica presso la Scuola Superiore Sant'Anna di Pisa, dove insegna Real-Time Systems e Neural Networks. Attualmente si occupa di architetture e algoritmi per fornire un supporto predicibile agli algoritmi di controllo e all'Intelligenza Artificiale.